MG4GPU STATUS

FOR CMS-MG JOINT MEETING

24.08.13

OVERVIEW

Gridpack Production

- ✓ Synced MG4GPU repo just after the last meeting (Aug 13th) - Regenerated numbers!
- ✓ Most slots in LXPLUS are in use🥲 - moved to the server in SNU for FORTRAN/CPP test
- ✓ Timing based on "nb_core" - i.e. no. of submitted madevents
  - ⇨ Synced nb_core with request_cpus in condor submission scripts
  - ⇨ 100% CPU usage for FORTRAN/CPP observed (based on htop)

Event generation

- ✓ Tested in LXPLUS - requesting single core exclusively
- ✓ Tested with no. of evts - 5K / 10K / 20K / 50K / 100K / 200K

GRIDPACK PRODUCTION

Drell-yan (low multiplicity)

	nb_core = 16 FORTRAN	nb_core = 16 CPP	nb_core = 16 CUDA
DY+0j	7m 15s	6m 29s	5m 21s
DY+1j	10m 13s	9m 59s	11m 39s
DY+2j	72m 10s	69m 49s	51m 14s
DY+012j	75m 48s	84m 42s	59m 36s

- First inclusive results - DY+2j dominates in gridpack production
- Compatible timing for FORTRAN ~ CPP (AVX2) - might expect more improvements in AVX512?

GRIDPACK PRODUCTION

Drell-yan (low multiplicity)

	FORTRAN	CPP	CUDA
DY+0j			
DY+1j			
DY+2j			
DY+012j			

- ✓ First inclusive resu
- ✓ Compatible timing

Results (full production)

	48 Intel	48 Intel, avx2
	FORTRAN	CPP
DY+0j	7m 59s	8m 38s
DY+1j	9m 27s	21m 3s
DY+2j	21m 24s	85m 6s
DY+3j	293m 38s	698m 41s
DY+4j	72 cms2 19362m 13s 13.5 days...	64 Intel, avx2 18509m 11s 12.8 days...

Previously it took longer...

GRIDPACK PRODUCTION

Drell-yan (low multiplicity)

	FORTRAN	CPP	CUDA
DY+0j	7m 15s	6m 29s	5m 21s
DY+1j	10m 13s	9m 59s	11m 39s
DY+2j	72m 10s	69m 49s	51m 14s
DY+012j	75m 48s	84m 42s	59m 36s

- First inclusive results - DY+2j dominates in gridpack production
- Compatible timing for FORTRAN ~ CPP (AVX2) - might expect more improvements in AVX512?
- Possible to find slots with AVX512 supports in LXPLUS condor - but hardly matchable:(

```
❯ condor_status -avail -const 'TotalCPUs == 48' -af Name TotalCPUs CPUs
slot1@b9g11p0067.cern.ch 48.0 1
slot1@b9g11p0080.cern.ch 48.0 1
slot1@b9g11p0183.cern.ch 48.0 1
slot1@b9g11p0258.cern.ch 48.0 1
slot1@b9g11p0279.cern.ch 48.0 2
slot1@b9g11p0432.cern.ch 48.0 1
slot1@b9g11p0544.cern.ch 48.0 1
slot1@b9g11p0887.cern.ch 48.0 1
slot1@b9g11p1207.cern.ch 48.0 6
slot1@b9g11p1287.cern.ch 48.0 12
```

Most cores are in use

GRIDPACK PRODUCTION

Drell-yan (high multiplicity)

	nb_core = 32 FORTRAN	nb_core = 32 CPP	nb_core = 16 CUDA
DY+3j	22h 39m	9h 4m	4h 18m
DY+4j	> 114h...(3% done)	> 103h...(6% done)	3d 22h
DY+01234j	> 113h...(2% done)	>89h...(5.5% done)	Not submitted yet

- Clearly(!) see the improvements in DY+3j, x2.5 for CPP and x8 for CUDA
- Question arises - What happens if I increase nb_core in CUDA? (Faster Production?)
  - ⇨ Possible to submit nb_core > total_threads (16), since threads are not fully used in CUDA case
  - ⇨ Memory exceeds in GPU side (2~3 GB for single madevent, 40 GB in A100)
- Possible to set-up in low-multiplicity cases (lower memory usage for single madevent)
- Better GPU? Multiple GPUs? - Reachable in LXPLUS

H100 IN LXPLUS

```
❯ condor_status -avail -const 'TotalCPUs == 48 && TotalGPUs == 1' -af Name TotalCPUs CPUs GPUs_DeviceName
slot1@b9pgpun005.cern.ch 48.0 48 NVIDIA H100 NVL
slot1@b9pgpun007.cern.ch 48.0 48 NVIDIA H100 NVL
slot1@b9pgpun008.cern.ch 48.0 48 NVIDIA H100 NVL
slot1@b9pgpun009.cern.ch 48.0 48 NVIDIA H100 NVL
slot1@b9pgpun010.cern.ch 48.0 48 NVIDIA H100 NVL
slot1@b9pgpun012.cern.ch 48.0 48 NVIDIA H100 NVL
❯ condor_status -avail -const 'TotalCPUs == 192 && TotalGPUs == 4' -af Name TotalCPUs CPUs GPUs_DeviceName
slot1@b9pgpun001.cern.ch 192.0 192 NVIDIA H100 NVL
slot1@b9pgpun002.cern.ch 192.0 192 NVIDIA H100 NVL
slot1@b9pgpun003.cern.ch 192.0 192 NVIDIA H100 NVL
slot1@b9pgpun004.cern.ch 192.0 128 NVIDIA H100 NVL
slot1@b9pgpun013.cern.ch 192.0 192 NVIDIA H100 NVL
slot1@b9pgpun014.cern.ch 192.0 192 NVIDIA H100 NVL
slot1@b9pgpun015.cern.ch 192.0 192 NVIDIA H100 NVL
slot1@b9pgpun016.cern.ch 192.0 192 NVIDIA H100 NVL
```

✓✓ Are these machines reachable? - Yes!

Example submit.jds

```
executable              = gridpack_generation.sh
jobbatchname            = TT3j_CKM_LO_5f_CUDA_H100
arguments               = $(ClusterId).$(ProcId)
output                  = job.$(ClusterId).$(ProcId).out
error                   = job.$(ClusterId).$(ProcId).err
log                     = job.$(ClusterId).$(ProcId).log
request_CPUs            = 48
request_GPUs            = 1
+requirements           = regexp("H100", TARGET.GPUs_DeviceName)
+JobFlavour             = "nextweek"
+MaxRuntime             = 1814400
+SingularityImage       = "/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el8:latest"
queue
```

H100 IN LXPLUS

```
Apptainer> nvidia-smi
Fri Aug 23 09:02:07 2024
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.90.07              Driver Version: 550.90.07      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA H100 NVL                Off |   00000000:06:00.0 Off |                    0 |
| N/A   34C    P0             61W /  400W |       4MiB /  95830MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

48 Intel cores + 1 H100 GPU

192 Intel cores + 4 H100 GPU

Can't find anyone using it😛

```
Apptainer> nvidia-smi
Fri Aug 23 12:56:21 2024
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.90.07              Driver Version: 550.90.07      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA H100 NVL                Off |   00000000:3A:00.0 Off |                    0 |
| N/A   60C    P0            299W /  400W |   86869MiB /  95830MiB |    100%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   1  NVIDIA H100 NVL                Off |   00000000:3B:00.0 Off |                    0 |
| N/A   36C    P0             61W /  400W |       4MiB /  95830MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   2  NVIDIA H100 NVL                Off |   00000000:AD:00.0 Off |                    0 |
| N/A   38C    P0             63W /  400W |       4MiB /  95830MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   3  NVIDIA H100 NVL                Off |   00000000:AE:00.0 Off |                    0 |
| N/A   38C    P0             64W /  400W |       4MiB /  95830MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
+-----------------------------------------------------------------------------------------+
```

GRIDPACK PRODUCTION

TTbar

	nb_core = 16 FORTRAN	nb_core = 16 CPP	nb_core = 16 CUDA	nb_core = 12 CUDA - H100
TT+0j	5m 47s	7m 15s	4m 41s	-
TT+1j	11m 8s	10m 43s	7m 7s	-
TT+2j	74m 52s	38m 25s	21m 47s	-
TT+3j	> 119h...(19%)	> 19h (6%)	nb_core = 6 8h 11m	4h 53m
TT+0123j	> 118h...(20%)	31h 6m	8h 24m	4h 52s

- Improvements observed throughout the whole processes - x2 for CPP / x3.5 for CUDA for TT+2j
- Expecting huge improvements in TT+3j/0123j!
- Only 6 madevents possible to be submitted for TT+3j/0123j - gg →ttxggg takes ~ 6GB GPU memory
- Additional test with 12 madevents using H100 (~ 96 GB)
- Super-fast gridpack production viable if multi-gpu supports available! (H100 x 4 ~ 384 GB)
- Madevents in TT+3j possesses 2~6 GB for GPU memory → could it be allocated dynamically?
  e.g. Check the remaining memory in GPU and submit the madevent...

GRIDPACK PRODUCTION

Summary

- ✓ Improvements from the latest numbers - now also viable in CPP
- ✓ Some processes (e.g. gg_ttxggg) takes huge amount of GPU memory
  - ⇨ Parallelization level (nb_core) resticted by (LargestMadeventMemory / TotalMemory)
  - ⇨ Room for improvement in inclusive processes - low multiplicity processes consume low memory, even though high multiplicity processes are time - consuming
  - ⇨ We don't have SUPER MEMORY SINGLE GPU possible to submit O(100) jobs...
    Viable for Super fast Gridpack Production if Multi-GPU setup supported!
- ✓ Experience with single H100 - x2 memory, ~x2 madevent jobs, ~/2 timing
- ✓ DY4j / TT3j too slow in FORTRAN/CPP set-up, many HPCs are in use:< - hard to produce numbers

EVENT GENERATION

- **Test Environment**
- ✓ Using single core(requesting 1 CPUs) in lxplus condor
- ✓ Test timing for 5k, 10k, 20k, 50k, 100k, 200k event generation
- ✓ Each process x nevt configuration tested 8 times - take avg & stddev

TT2j - 5000

	FORTRAN		CPP		CUDA	
0	28m9.804s	1689.804	20m38.510s	1238.51	6m47.389s	407.389
1	29m8.745s	1748.745	20m2.217s	1202.217	7m45.196s	465.196
2	28m45.357s	1725.357	19m54.762s	1194.762	6m53.662s	413.662
3	27m32.752s	1652.752	19m52.637s	1192.637	7m50.752s	470.752
4	28m22.442s	1702.442	21m0.456s	1260.456	6m18.169s	378.169
5	27m29.009s	1649.009	19m47.381s	1187.381	6m43.447s	403.447
6	27m21.541s	1641.541	20m11.514s	1211.514	6m0.925s	360.925
7	28m47.916s	1727.916	20m31.408s	1231.408	6m19.430s	379.43
avg		1692.196		1214.861		409.871
err		40.832		26.011		19.770

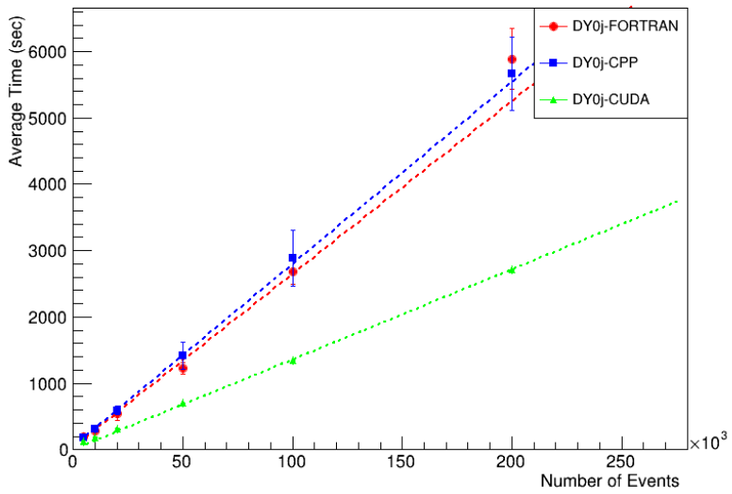
These numbers used for results

- ✓ Done for DY+0j/1j, TT+0j/1j/2j, Partially done for DY+2j, DY+012j

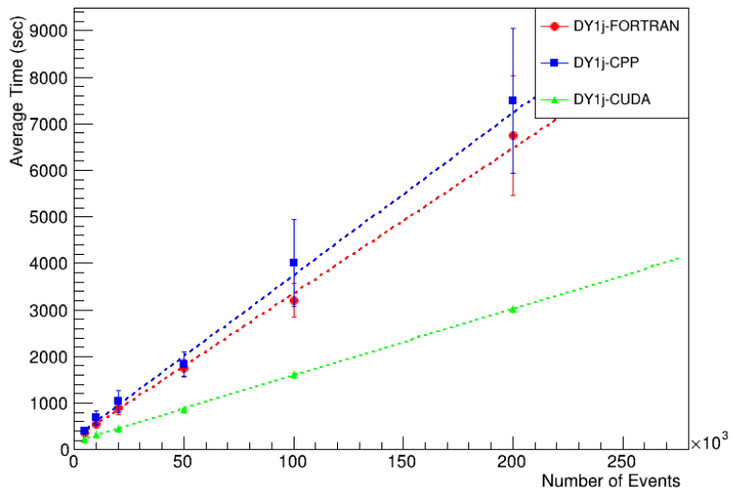
EVENT GENERATION

Drell-Yan

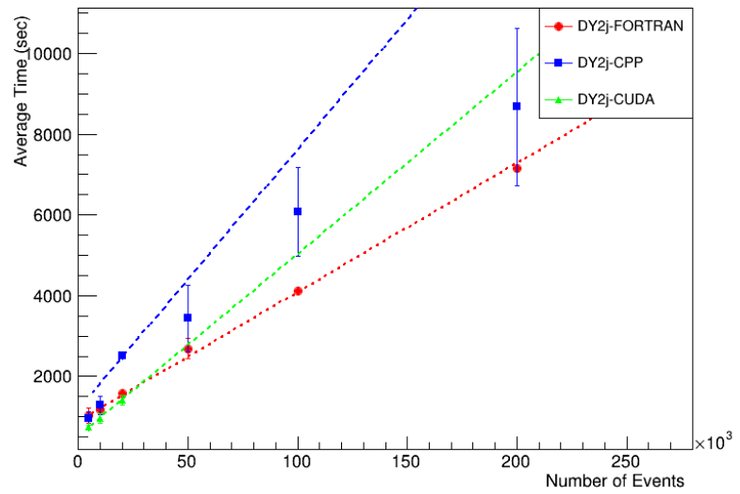
DY+0j



DY+1j



DY+2j



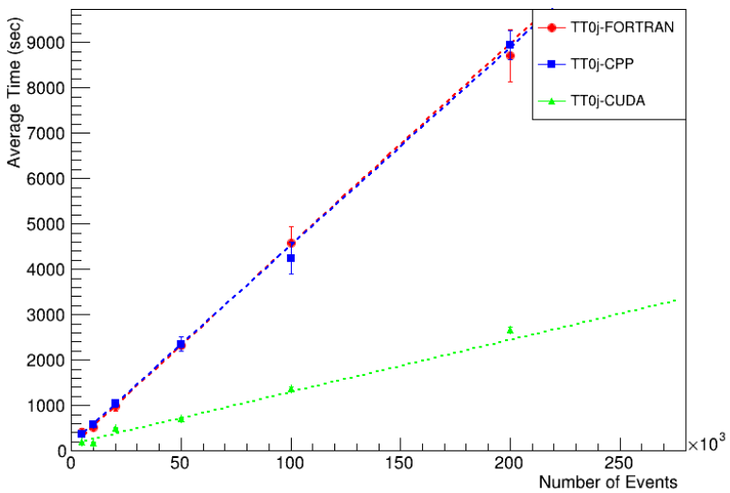
CUDA done for 5k, 10k, 20k only

✓ Clearly see x2-3 improvement in CUDA! No improvement viable for AVX2 Vectorization

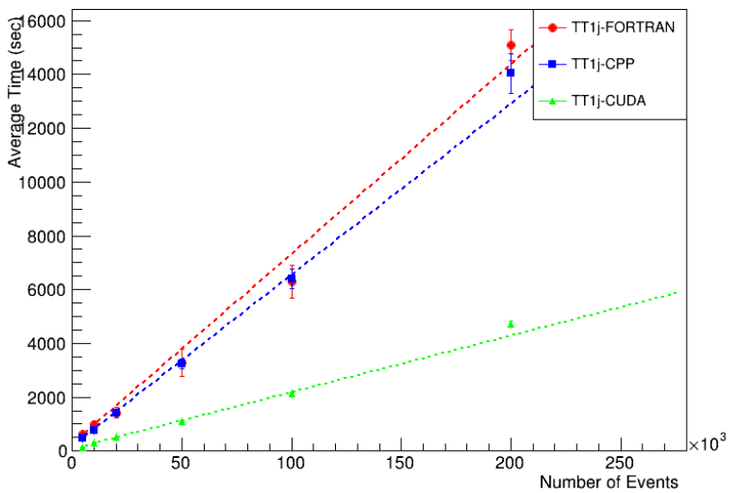
EVENT GENERATION

TTbar

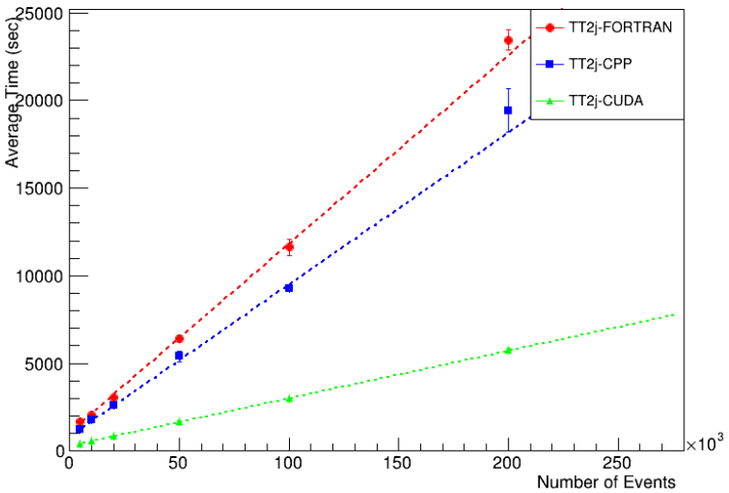
TT+0j



TT+1j



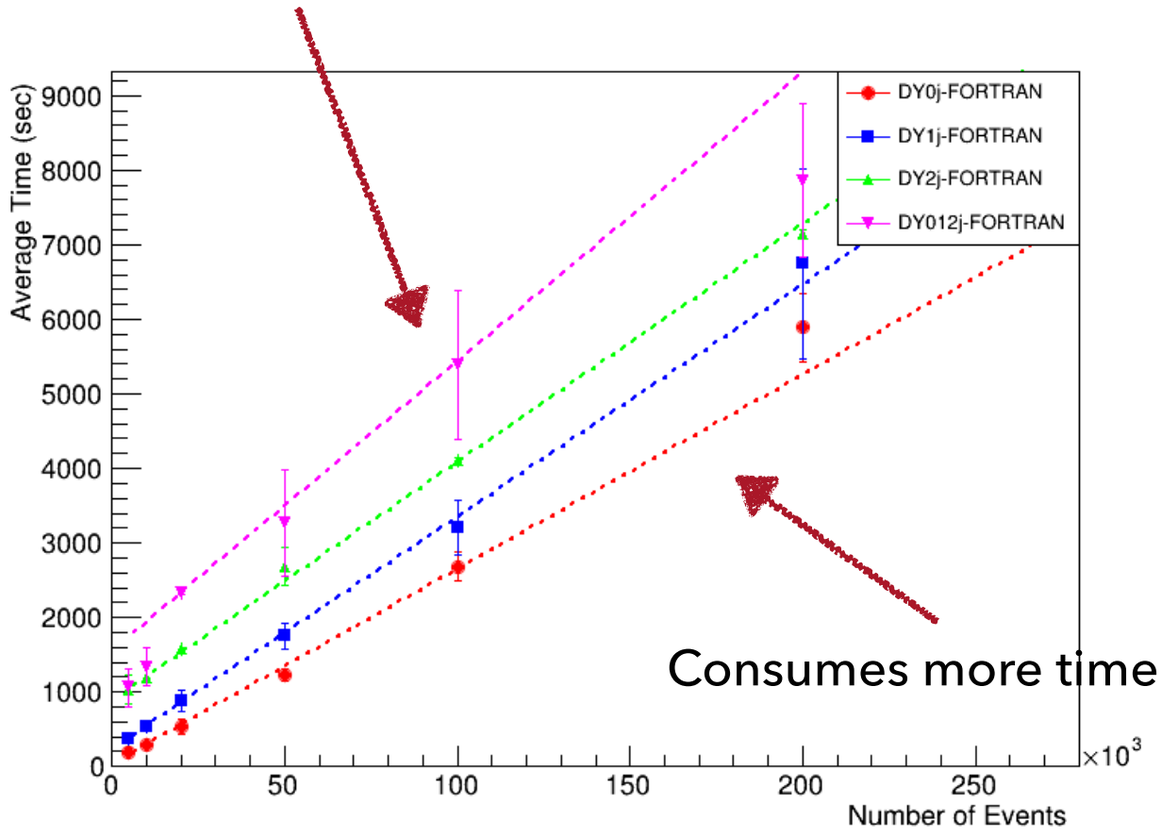
TT+2j



- x4 improvement in CUDA. SOME improvement viable for AVX2 vectorization... but not large.
- Improve increases with final state multiplicity...Will be checked with TT+3j

Partial results with inclusive sample

Large fluctuation & Not linearly scalable?



Consumes more time than exclusive samples

EVENT GENERATION

Summary

- Clear improvement (x2-4, depending on the process) shown in CUDA!
  Not large (or no) improvement for AVX2 supports....
- AVX512 Supports? Again, lack of machines.
- DY+012j inclusive sample takes more time than DY+0j/1j/2j
  - Comparison flamegraphs for DY+2j / DY+012j would help
- Further test with higher multiplicities are on-going